SPIKE PRIME LESSONS

By the Creators of EV3Lessons

# MICROPYTHON ON SPIKE PRIME

BY SANJAY AND ARVIND SESHAN

# LESSON OBJECTIVES

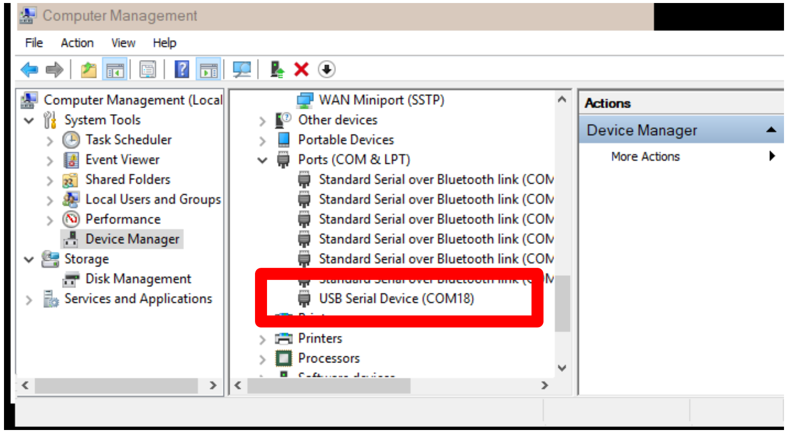
- Learn how to use MicroPython REPL on SPIKE Prime

- To create full programs in VS Code to run on the hub follow the instructions at https://github.com/sanjayseshan/spikeprime-vscode/wiki

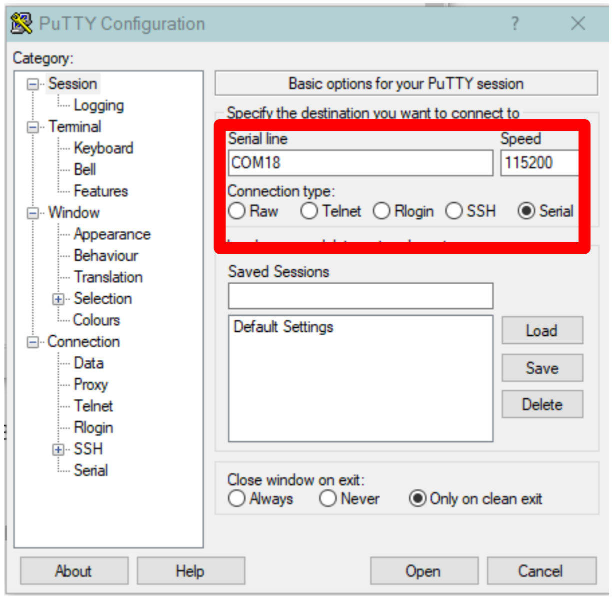
# STEP 1: CONNECT (WINDOWS)

- Install any terminal emulator of your choice
- Example: PuTTY [https://www.putty.org/](https://www.putty.org/)

- Make sure your SPIKE Prime software is not running
- Connect the Hub via USB port to your computer
- Find the port
  - On a PC, look in your device manager (in start menu→ Windows Administrative Tools→ Computer Management→ Device Manager) under serial to see what serial ports you have connected
  - If you have multiple USB serial ports try disconnecting and reconnecting to see which one appears
- Connect to the right port at 115200 baud



USB Serial Device (COM18)



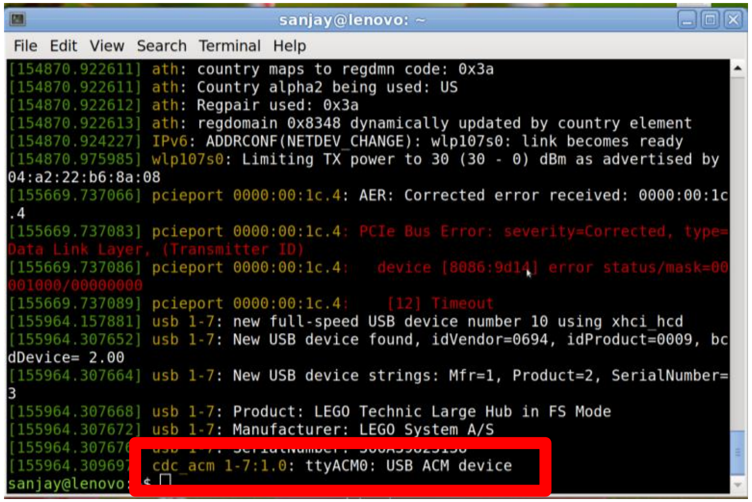
Serial line: COM18

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

# STEP 1: CONNECT (DEBIAN GNU/LINUX)

1. Open terminal – This can be found in Applications → System Tools usually
2. Type in the following commands (this is for Debian and derivatives):
   1. `sudo apt-get update`
   2. `sudo apt-get install -y screen`
3. Plug in your hub and run `sudo dmesg`. A long list of log messages will show up. The last line(or close to last) should contain USB ACM device and the id similar to ttyACM0. If you cannot find it, first look for LEGO Technic Large Hub.
4. Run `sudo screen /dev/ttyACM0 115200`. Replace ttyACM0 with your id.



```
[154870.922611] ath: country maps to regdmn code: 0x3a
[154870.922611] ath: Country alpha2 being used: US
[154870.922612] ath: Regpair used: 0x3a
[154870.922613] ath: regdomain 0x8348 dynamically updated by country element
[154870.924227] IPv6: ADDRCONF(NETDEV_CHANGE): wlp107s0: link becomes ready
[154870.975985] wlp107s0: Limiting TX power to 30 (30 - 0) dBm as advertised by
04:a2:22:b6:8a:08
[155669.737066] pcieport 0000:00:1c.4: AER: Corrected error received: 0000:00:1c
.4
[155669.737083] pcieport 0000:00:1c.4: PCIe Bus Error: severity=Corrected, type=
Data Link Layer, (Transmitter ID)
[155669.737086] pcieport 0000:00:1c.4:   device [8086:9d14] error status/mask=00
001000/00000000
[155669.737089] pcieport 0000:00:1c.4:    [12] Timeout
[155964.157881] usb 1-7: new full-speed USB device number 10 using xhci_hcd
[155964.307652] usb 1-7: New USB device found, idVendor=0694, idProduct=0009, bc
dDevice= 2.00
[155964.307664] usb 1-7: New USB device strings: Mfr=1, Product=2, SerialNumber=
3
[155964.307668] usb 1-7: Product: LEGO Technic Large Hub in FS Mode
[155964.307672] usb 1-7: Manufacturer: LEGO System A/S
[155964.30767[illegible] usb 1-7: SerialNumber: [illegible]
[155964.30969[illegible] cdc_acm 1-7:1.0: ttyACM0: USB ACM device
sanjay@lenovo:~$ 
```

# STEP 1: CONNECT (MAC OS X)

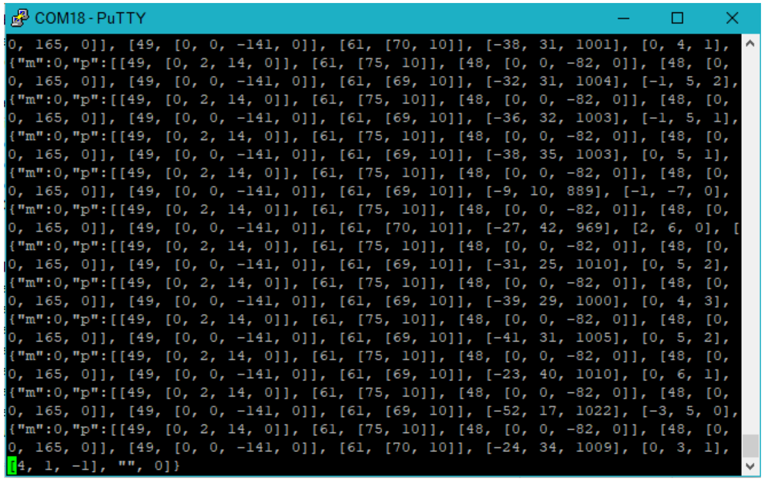
1. Open terminal – This can be found in Applications → Utilities usually
2. Run `ls /dev/ | fgrep usb | fgrep tty` to find the hub port
3. Run `screen /dev/tty.usbmode366A398231381 115200`. Replace tty… with the output of the first command. If there were multiple outputs, try all of them until you get the right one.

Below is an example of the commands being run. The computer output is in green, the commands you type are in black.

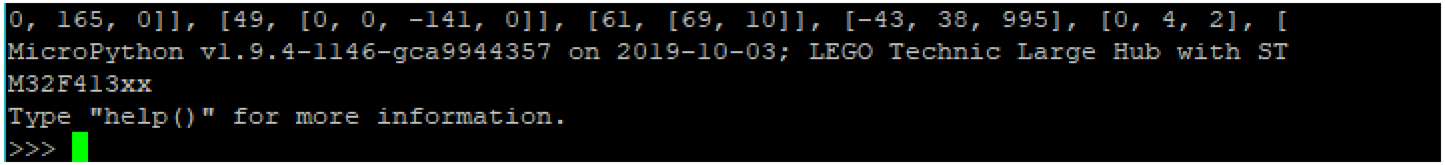
```
$ ls /dev | fgrep usb | fgrep tty
tty.usbmodem366A39831234
$ screen /dev/tty.usbmodem366A39831234
```

# STEP 2

- You may see a lot of numbers scroll by. This is the sensor and motor log.
- Hit Control-C
- You will be ready to program



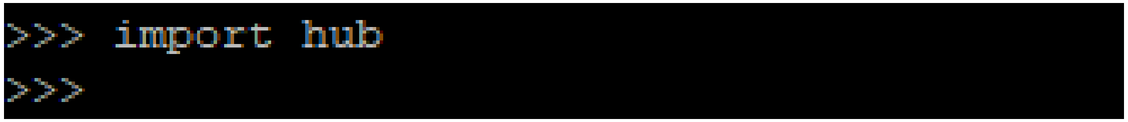
```
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [70, 10]], [-38, 31, 1001], [0, 4, 1],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-32, 31, 1004], [-1, 5, 2],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-36, 32, 1003], [-1, 5, 1],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-38, 35, 1003], [0, 5, 1],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-9, 10, 889], [-1, -7, 0],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [70, 10]], [-27, 42, 969], [2, 6, 0], [
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-31, 25, 1010], [0, 5, 2],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-39, 29, 1000], [0, 4, 3],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-41, 31, 1005], [0, 5, 2],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-23, 40, 1010], [0, 6, 1],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-52, 17, 1022], [-3, 5, 0],
{"m":0,"p":[[49, [0, 2, 14, 0]], [61, [75, 10]], [48, [0, 0, -82, 0]], [48, [0,
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [70, 10]], [-24, 34, 1009], [0, 3, 1],
[4, 1, -1], "", 0]}
```



```
0, 165, 0]], [49, [0, 0, -141, 0]], [61, [69, 10]], [-43, 38, 995], [0, 4, 2], [
MicroPython v1.9.4-1146-gca9944357 on 2019-10-03; LEGO Technic Large Hub with ST
M32F413xx
Type "help()" for more information.
>>>
```

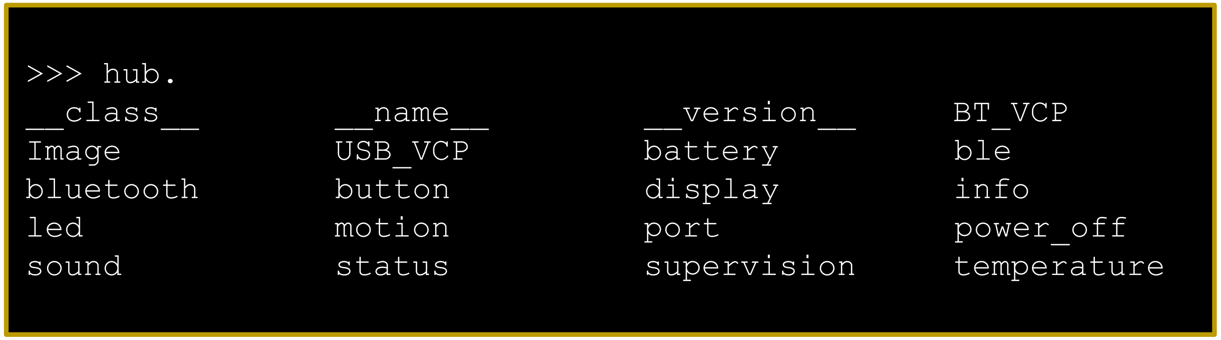
# HUB MODULE

- The "hub" python module contains all the key functions/objects that are necessary to interact with the SPIKE Prime hub.
- To access this module, you must first "import" the module. Type "import hub" at the MicroPython prompt



```
>>> import hub
>>>
```

- Once you have imported hub, you can explore some of the interfaces it exposes using autocompletion. Type "hub." (make sure to include the period) and then hit the "Tab" button



```
>>> hub.
__class__       __name__        __version__     BT_VCP
Image           USB_VCP         battery         ble
bluetooth       button          display         info
led             motion          port            power_off
sound           status          supervision     temperature
```

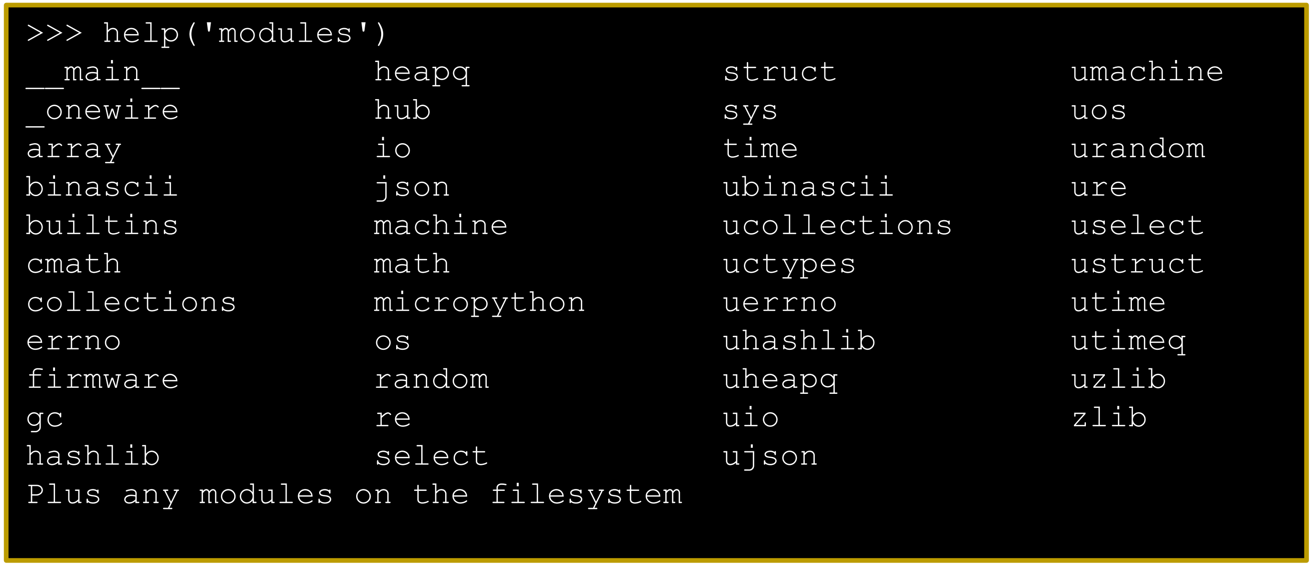
# HELP COMMAND

- The on-brick MicroPython also provides a limited help tool. To access help, type "help()"

```
>>> help()
Welcome to MicroPython!

For online help please visit http://micropython.org/help/.

Quick overview of commands for the board:
  hub.info()     -- print some general information
  hub.status()   -- print sensor data

Control commands:
  CTRL-A         -- on a blank line, enter raw REPL mode
  CTRL-B         -- on a blank line, enter normal REPL mode
  CTRL-C         -- interrupt a running program
  CTRL-D         -- on a blank line, do a soft reset of the board
  CTRL-E         -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)
For a list of available modules, type help('modules')
```

# OTHER MODULES/LIBRARIES

- Type the `help(‘modules‘)` command (make sure to type the quotes)

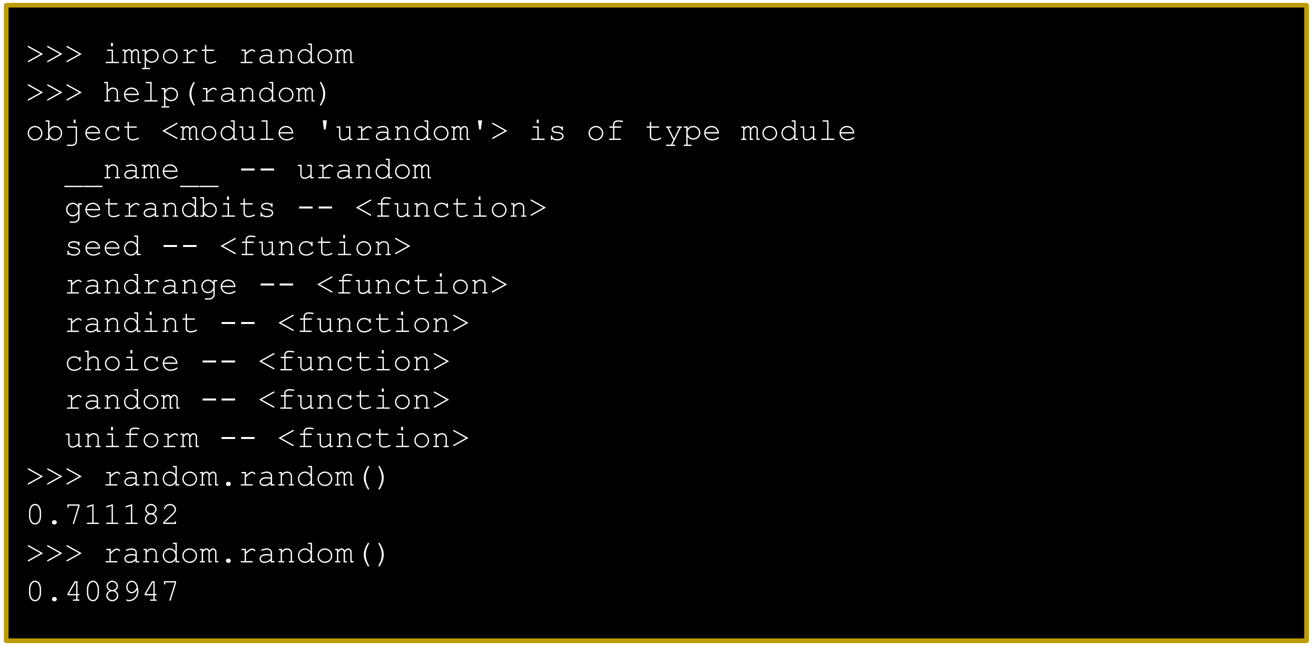


```
>>> help('modules')
__main__          heapq             struct            umachine
_onewire          hub               sys               uos
array             io                time              urandom
binascii          json              ubinascii         ure
builtins          machine           ucollections      uselect
cmath             math              uctypes           ustruct
collections       micropython       uerrno            utime
errno             os                uhashlib          utimeq
firmware          random            uheapq            uzlib
gc                re                uio               zlib
hashlib           select            ujson
Plus any modules on the filesystem
```

- This provides a list of modules that are available on the SPIKE Prime

# OTHER MODULES/LIBRARIES

- You can use the import command to load any of the libraries you find and then use autocompletion or help() to explore their functions



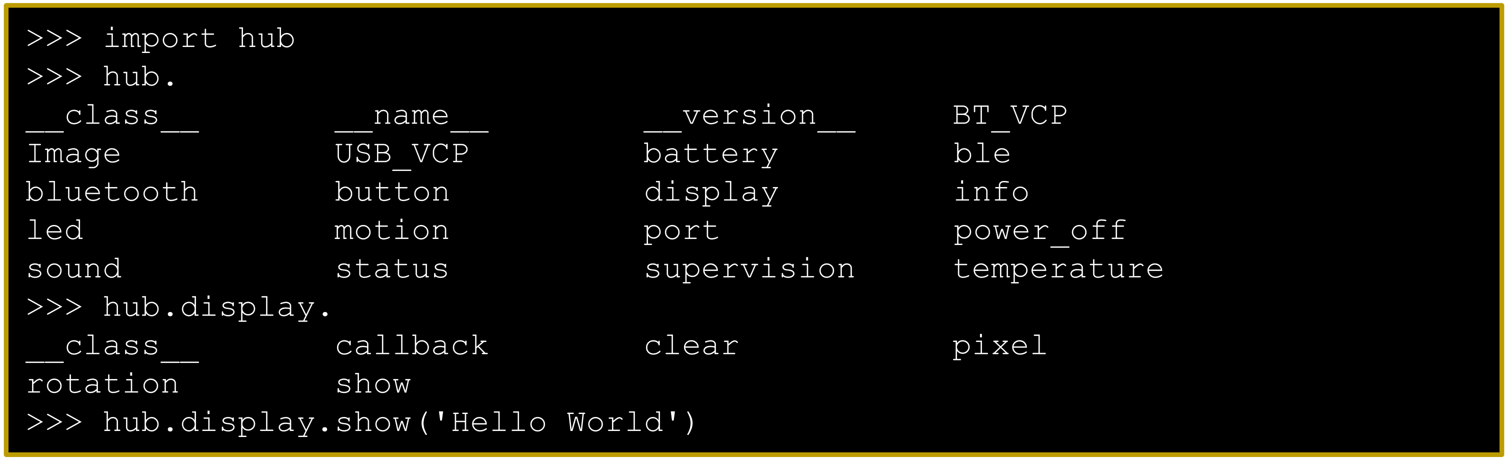
```
>>> import random
>>> help(random)
object <module 'urandom'> is of type module
  __name__ -- urandom
  getrandbits -- <function>
  seed -- <function>
  randrange -- <function>
  randint -- <function>
  choice -- <function>
  random -- <function>
  uniform -- <function>
>>> random.random()
0.711182
>>> random.random()
0.408947
```

# CHALLENGE 1: HELLO WORLD

- Print "Hello World" on your Hub Light Matrix
- Some key steps:
  1. Import the hub module
  2. Explore the components of the hub to find one that controls the Light Matrix (hint: you want to "display" something)
  3. Finally, look for a method that "shows" something on the display

# CHALLENGE 1: HELLO WORLD

- Print “Hello World” on your Hub Light Matrix



```
>>> import hub
>>> hub.
__class__       __name__        __version__     BT_VCP
Image           USB_VCP         battery         ble
bluetooth       button          display         info
led             motion          port            power_off
sound           status          supervision     temperature
>>> hub.display.
__class__       callback        clear           pixel
rotation        show
>>> hub.display.show('Hello World')
```

# CREDITS

- This lesson was created by Sanjay Seshan and Arvind Seshan for SPIKE Prime Lessons
- More lessons are available at www.primelessons.org